REMARKS

Claims 11-29, 43-64, and 69-93 were pending. Claims 72 and 83 have been amended. No claims have been added/canceled. Accordingly, claims 11-29, 43-64, and 69-93 remain pending subsequent entry of the present amendment.

35 U.S.C. § 112 Rejections

Claims 72 and 74-93 stand rejected under 35 U.S.C. § 112, second paragraph. Claims 72 and 83 have been amended in a manner believed to overcome the rejection. Accordingly, dependent claims 74-93 are believed to overcome the rejection.

35 U.S.C. § 102 Rejections

Claims 11-29, 43-64, 70-72 and 74-93 stand rejected under 35 U.S.C. § 102(e) as being anticipated by newly cited U.S. Patent No. 6,892,282 (hereinafter "Hass"). Claims 72 and 74-93 stand rejected under 35 U.S.C. § 102(b) as being anticipated by newly cited U.S. Patent No. 6,112,283 (hereinafter "Neiger"). Applicant respectfully traverses these rejections and requests reconsideration in view of the following comments.

Claim 11 recites a memory controller coupled to a memory, to a plurality of microprocessors, and to a global arbiter, wherein the global arbiter is configured to:

> "receive one or more requests for a memory transaction;
> <u>assign a global order to each of the one or more requests;</u>
> <u>execute according to the global order, a snoop corresponding to each request</u>, wherein each snoop comprises determining whether one or more of the plurality of microprocessors has data associated with a corresponding request; and
> <u>respond to each request according to the global order</u>." (emphasis added)

As can be seen in the highlighted features, the global arbiter of the claimed invention is configured to "assign a global order to each of the one or more requests". Applicant has reviewed the cited reference and submits Hass nowhere discloses these features. For example, Hass discloses the functions of a global snoop controller 22, but nowhere does Hass disclose an assignment of a global order to each of the one or more memory requests. Hass discloses the functions in the following:

> "For purposes of illustration, FIG. 2 demonstrates the process with cluster 12—the same process is applicable to clusters 14, 16, and 18.
>
> Processing cluster 12 determines whether a memory location for an application operation is mapped into the cache memory in cluster 12 (step 30). If cluster 12 has the location, then cluster 12 performs the operation (step 32). <u>Otherwise, cluster 12 issues a request for the necessary memory location to global snoop controller 22 (step 34).</u> In one embodiment, cluster 12 issues the request via point-to-point connection 13. As part of the request, cluster 12 forwards a request descriptor that instructs snoop controller 22 and aids in tracking a response to the request.
>
> <u>Global snoop controller 22 responds to the memory request by issuing a snoop request to clusters 14, 16, and 18 (step 36)</u>. The snoop request instructs each cluster to transfer either ownership of the requested memory location or the location's content to cluster 12. Clusters 14, 16, and 18 each respond to the snoop request by performing the requested action or indicating it does not possess the requested location (step 37). In one embodiment, global snoop controller 22 issues the request via snoop ring 21, and clusters 14, 16, and 18 perform requested ownership and data transfers via snoop ring 21…
>
> <u>Global snoop controller 22 analyzes the clusters' snoop responses to determine whether the snooped clusters owned and transferred the desired memory (step 38).</u> If cluster 12 obtained access to the requested memory location in response to the snoop request, cluster 12 performs the application operations (step 32). Otherwise, global snoop controller 22 instructs EBL 24 to carry out an access to main memory 26 (step 40)." (Hass, col. 5 lines 21-64) (emphasis added)

As with the remainder of the document, this citation does not disclose the global snoop controller 22 performs the features of the claimed invention such as assigning a global order, executing a snoop according to the global order, and responding to each request according to the global order. As stated above, the "[g]lobal snoop controller 22 responds to the memory request by issuing a snoop request to clusters 14, 16, and 18 (step 36)." However, there is no mention of assigning a global order to the requests, or issuing the requests to the clusters in a global order.

The Examiner has indicated that the second tier cache of Hass includes these features as described below. In the present Office Action it states that the "second tier cache functions of Hass may be considered a part of the functions of the claimed global arbiter (along with those of the global snoop controller of Hass). The second tier cache receives requests and orders those requests via the input queues shown in Fig. 6. Since snoops for those requests are then placed on the snoop ring by the global snoop controller, those snoops remain in the order established for the requests, and thus the response are also in the same order." However, even if the functionality of the second tier caches 80 are considered part of the functions of the global snoop controller 22, there is no assigning a global order to each of the one or more requests, executing a snoop according to the global order, and responding to each request according to the global order. A second tier cache 80 assigns priority of memory requests within its own corresponding processing cluster 12. The priority is not global. Further, the memory requests are local to the corresponding cluster and only the snoop requests are global. The snoops that arrive via snoop ring 21, as shown above, are not assigned a global order by the global snoop controller 22 and these snoops are not assigned a global order by the second tier cache 80. Rather, these snoops are assigned a higher priority over the local cluster's first tier caches such as within processing cluster 12 of FIG. 4. Hass discloses these features in the following:

> "FIG. 6 illustrates a pipeline of operations implemented by second tier cache 80 in one embodiment of the present invention. <u>In stage 370, cache 80 accepts memory requests. In one embodiment, cache 80 is coupled to receive memory requests from external sources</u> (Fill),

> global snoop controller 22 (Snoop), first tier data caches 52, 92, 96, and 100 (FTD-52; FTD-92; FTD-96; FTD-100), and first tier instruction caches 54, 94, 98, and 102 (FTI-54; FTI-94; FTI-98; FTI-102) …
>
> Cache 80 also includes snoop queue 390 for receiving and maintaining requests from snoop ring 21. Upon receiving a snoop request, cache 80 buffers the request in queue 390 and forwards the request to the next cluster on snoop ring 21." (Hass, col. 11 line 41 – col. 12 line 2) (emphasis added)
>
> Cache 80 includes a scheduler for assigning priority to the above-described memory requests. In stage 370, the scheduler begins the prioritization process by selecting requests that originate from snoop queue 390 and each of compute engines 50, 86, 88, and 90, if any exist. For snoop request queue 390, the scheduler selects the first request with a Validity field showing the request is valid. In one embodiment, the scheduler also selects an entry before it remains in queue 390 for a predetermined period of time.
>
> For each compute engine, the scheduler gives first tier instruction cache requests (FTI) priority over first tier data cache requests (FTD). In each data cache request queue (382, 384, 386, and 388), the scheduler assigns priority to memory requests based on predetermined criteria. In one embodiment, the predetermined criteria are programmable." (Hass, col. 12 lines 32-46)

As can be seen from the above, second tier cache 80 buffers a snoop request and forwards it to the next cluster on snoop ring 21. No global order is assigned to it. Hass nowhere teaches that the arrival of the snoop request determined by global snoop controller 22 is based on an assigned global order. Also, second tier cache 80 selects the first snoop request with a Validity field showing the request is valid. There is no selection logic to select a snoop request based on a global order. For all of the reasons above, claim 11 is patently distinct from the cited art.

As independent claims 43, 53, 61and 72 include features similar to claim 11, claims 43, 53, 61and 72 are patentably distinguished from the cited reference for similar reasons. As each of the dependent claims include the features of the independent claims

on which they depend, each of the dependent claims are patentably distinct for at least the above reasons.

In addition, claim 72 recites a memory controller configured to:

> "receive one or more memory requests;
> assign a global order to each of the one or more requests;
> execute according to the global order, a snoop corresponding to each request, wherein each snoop comprises determining whether one or more of the plurality of agents has data associated with a corresponding request; and
> respond to each request according to the global order;
> ordering logic for establishing a global order for said requests, and for insuring that said initiated snoops conform to said global order."

Neiger teaches away from the features of claim 72 such as "receive one or more memory requests". Rather, Neiger discloses a computer system comprising processing circuitry for receiving snoop requests as in the following:

> "a computer system includes nodes connected through conductors. At least some of the nodes each include memory and processing circuitry <u>to receive snoop requests in a node reception order and to initiate snoops of the memory in the node before the snoop requests are in a global order</u>." (Neiger, col. 1 lines 56-59) (emphasis added)
>
> "Referring to FIG. 1, an exemplary computer system 10 includes nodes N0, N1, N2, and N3. The nodes communicate with each other through a point-to-point ring topology rather than a shared bus … For example, virtual slots, described below, may be transmitted on only some of the conductors of conductors 22-28. <u>Snoop responses may be transmitted on different conductors than snoop requests</u>. In some embodiments, some signals, such as data signals, are sent on conductors not included in conductors 22-28." (Neiger, col. 2 lines 39-56). (emphasis added).
>
> "FIG. 4 illustrates certain circuitry included in an exemplary embodiment of control circuitry 40 of node N2 of FIG. 3 … Processing circuitry 66 may performs various functions such as providing messages to processor 39. In some embodiments, <u>processing</u>

> circuitry 66 initiate a snoop of memory, such as cache 34." (Neiger, col. 6 line 66 – col. 7 line 7). (emphasis added).

Applicant has reviewed the cited reference and submits nowhere in Neiger is it disclosed that the receiving circuitry 42 or control circuitry 40 receive memory requests, but rather only snoop requests as taught in the above citations. Additionally, Neiger does not disclose assigning a global order to each of the one or more memory requests. For at least these reasons, claim 72 is patently distinct from the cited art.

Furthermore, claim 72 recites the memory controller is configured to "execute according to the global order, a snoop corresponding to each request". In contrast, Neiger teaches out-of-order snooping as in the following:

> "Some embodiments of the invention allow initiation of snooping upon receipt of the snoop request, while still maintaining ultimate cache coherence. The invention allows initiation of snooping prior to completion of ordering of the requests.
>
> The detailed technique for out-of-order snooping described below involves properties 1-5 recited above. However, properties 4 and 5 are not required.
>
> It is permissible to perform snoops in node reception order as long as results (both snoop results and writebacks) are distributed in global order. Changes to the tags may be those that would occur had the snoops been performed in global order." (Neiger, col. 10 lines 47-59).

In addition, Neiger nowhere discloses "ordering logic for … insuring initiation of said snoops conform to said global order". As Neiger does not perform these features, it is obvious that Neiger does not disclose ordering logic for "insuring initiation of said snoops conform to said global order". For these further reasons, claim 72 is patently distinct from the cited reference.

Finally, Applicant traverses the rejections of paragraph 9 which suggest all the features of claims 13-29, 49 and 51 are either disclosed or inherent in Hass. If such

features are disclosed, Applicant requests a citation to such disclosures. Additionally, Applicant submits such features are not inherent as inherency requires such features be present in Hass. For example, there is no requirement that a request to share results from a read miss in a cache (claim 14) must be present in Hass.

Applicant believes all claims to be in condition for allowance and withdrawal of the rejections is requested.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

Respectfully submitted,

/James W. Huffman/

James W. Huffman
Reg. No. 35,549
ATTORNEY FOR APPLICANT(S)

Date: 1/18/08